Assignment 3
Vision and Image Processing

Fredrik Holsten Tobias Ludwig

January 8, 2018

Introduction

Stereo Correspondence Analysis (in contrast to Photometric Stereo) tries to infer depth by horizontal displacement of objects between two images taken from different camera positions. Problems of this approach are that there might be occlusions and objects that leave the image at the boundaries.

To get the displacement, dense matching of the images can be employed. We are doing it in a scale space, which is an efficient approach because it minimizes the search space.

In the end, we will run our implementation on Tsukuba's and the Venus data set and compare the results with the given ground truth disparities.

Matching and enhancement

The given pairs of pictures are already aligned in height, so corresponding points lie on epipolar lines. An image can therefore be matched row-wise (with the patch size as row height). We apply the whole matching procedure twice, from left to right and from right to left. Ideally, this would give us two disparity maps, mapL and mapR, with the following attributes:

$$mapL[i, j + mapR[i, j]] - mapR[i, j] == 0 \quad \forall (i, j) \in [1..m, 1..n] \tag{1}$$
$$mapR[i, j + mapL[i, j]] - mapL[i, j] == 0 \quad \forall (i, j) \in [1..m, 1..n] \tag{2}$$
$$\tag{3}$$

where (m,n) is the shape of the maps. In other words, if the left-to-right disparity map tells us to look 3 units to the right, then the right-to-left map should tell us, in the displaced position, to look 3 units to the left, and vice versa. This property can be used for mutual correction in two-way matching. We implemented this, but did not see much difference in the result, so we ended up not using it. This can be due to the fact that we used a median filter to correct for outlier disparities, which inevitably undermines the effectiveness of two-way matching.

We can smooth the disparity maps with a median filter under the assumption that the surfaces are piece wise smooth. For instance, a 1 surrounded by 5's is most likely supposed to be a five. Another thing we do, is to demand that the normalized cross correlation between to patches is at least 0.2. If it is bellow this, we regard it safe to assume that the patches does not belong together.

Patch matching

We use normalized cross correlation to identify similar patches. NCC is displacement measure in the range $[-1, 1]$ where 1 means perfect alignment. We can find matching patches rather effectively by correlating one patch from one image over the whole search region in the other. The index of the peak in the normalized cross correlation is translated to the appropriate index in the right image. Skimage provides the function `match_template` that does just that, so we use that. An easier version of the algorithm is given in source code by (Solem, 2012, p.50).

Pyramid

Using an image pyramid gives us gross information where to search for matches in finer levels, thereby reducing the search space. We start with the image itself and keep downscaling it by a constant factor (default 2) for 3 levels. The smallest level is matched and the resulting disparity map is propagated up. On the next level we can use it (after upsampling the values) as a prior for the corresponding pixel locations in the other image. As we move down the scale pyramid, we can detect finer structures, such as the stick of the lamp in Tsukuba's image. We account for those small/thin things not visible at the coarse scale but moving at the finer scale by leaving a search radius $\pm M$ which is another parameter of the algorithm (along with the patch size N).

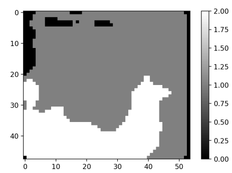
Results

The image sets Venus and Map comes with a ground truth disparity map. We used this to evaluate our implementation by computing mean error, the standard deviation of the error, as well as the number disparities that deviate more than 3 pixels from the ground truth.

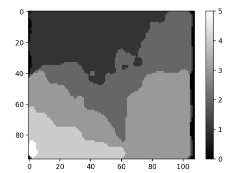
Venus

Figure 1 shows how we narrow down the real disparities for Venus.

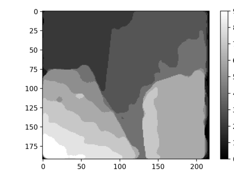
We see that in the fourth level, only rough features are made out. As we go up in the image pyramid, we see that these features are getting more nuanced. Let's investigate how the mean error changes as we change the search radius M and patch size N. As we see from table 1, the mean error appears to decrease as we increase the patch size. This makes intuitive sense, since a larger patch size



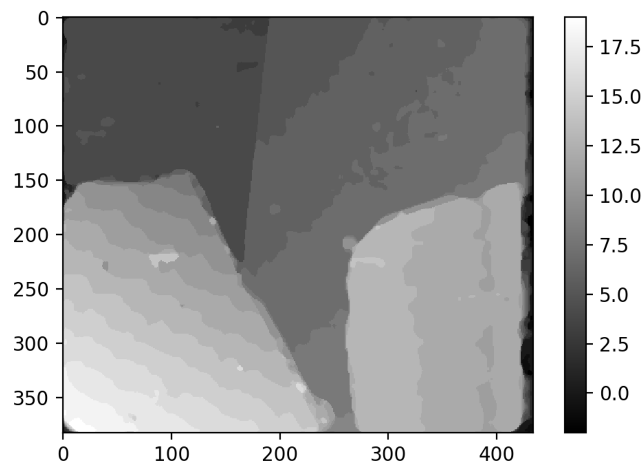
(a) Fourth level



(b) Third level



(c) Second level



(d) First level

Figure 1: Disparity pyramid for Venus

means a lower probability of mismatches. Before reviewing the search radius' impact on the mean error, it should be noted that this is a parameter that has to be set with respect to the images in question. It should not be lower than the largest displacement in the two images. We see that the mean error is lowest for M = 2, and larger for M=3 and 4. If the largest displacement in the image was larger, then an optimal M would probably have been larger than 2.

Table 2 lists the standard deviation of the error. A high standard deviation implies that the algorithm performs better in some areas than others. We see that as the patch size increases, the standard deviation does too. When we increase the search radius on the other hand, the standard deviation decreases. This is very much in line with the results from table 1: a larger patch size gives more stable results, whereas M has to be fitted to the images in question.

Table 3 shows the number of large errors (deviations larger than 3 pixels). Surprisingly, we see that they are more frequent when we increase the patch size. Still these numbers are all quite acceptable when you divide them by the number of total pixels in the image (434x383) these translate to an error rate

Table 1: Mean error (pixels), Venus

M \N	5	7	9	11
2	-0.3493	-0.1198	-0.0741	-0.0291
3	-0.2432	-0.1523	-0.1179	-0.0631
4	-0.3525	-0.0938	-0.0635	-0.0608

Table 2: Standard deviation of error, Venus

M \N	5	7	9	11
2	2.4068	1.8472	1.8342	1.8054
3	2.2175	1.9430	1.9434	1.8171
4	3.2297	2.3307	2.2092	1.9064

Table 3: Number of large errors, Venus

M \N	5	7	9	11
2	4232	4710	5390	6258
3	4347	4628	5080	5652
4	5019	5453	5588	5522

Table 4: Statistics: Map

Mean error	Standard deviation of error	Number of large errors
1.3888	8.3539	9045

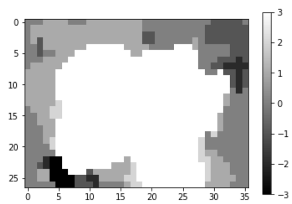
between 2,6% and 3,8%.

Map

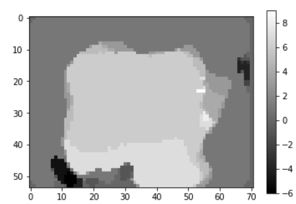
In figure 2, we can see the disparity pyramid for Map. There are really only two depth levels in this image set. The foreground and the background. We see clear distinction between the two, but the disparity surface suffers from some very noisy regions. These come from occluded areas, but also nearby the borders and the lower part of the foreground. The method of matching patches require that the surfaces have some texture. The map images have texture, but maybe the texture is too homogeneous. That the performance is not as good as with Venus can be seen in table 4 as well.

Tsukuba

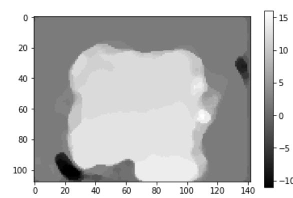
Our best Tsukuba result (Fig. 3) with parameters $N = 11$ and $M = 4$ is subjectively very close to the true disparity map shown in Fig. 4. The major 5 levels of depths (lamp, statue, desk, camera, background) are nicely distinguishable. The contours are not very sharp though which might be due to the big patch



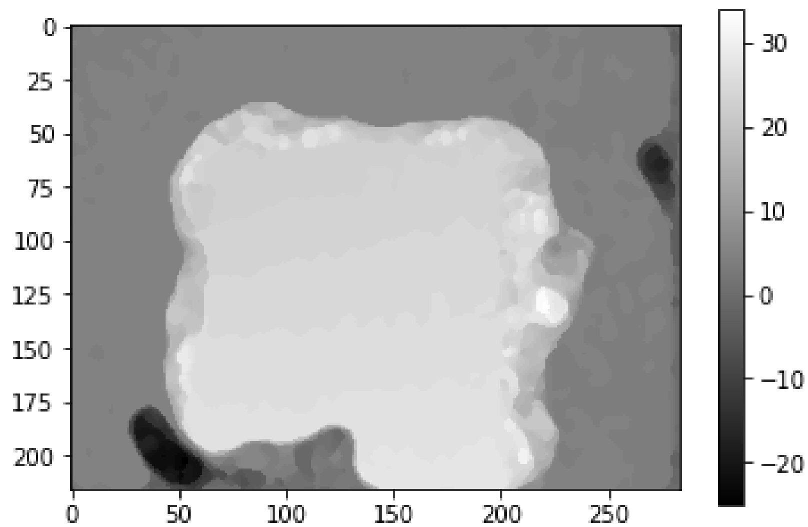
(a) Fourth level



(b) Third level



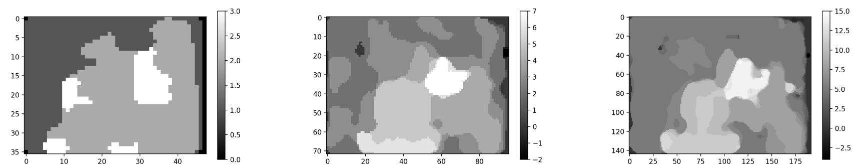
(c) Second level



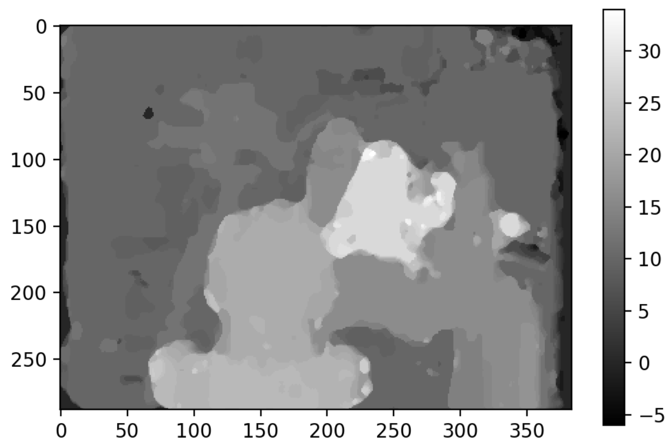
(d) First level

Figure 2: Disparity pyramid for Map

size and the median filter. Details like the arm of the lamp are lost because they were too thin. There is little depth information in the face. Some dark artifacts in the background (e.g. left from the eye of the camera) appear due to noise propagated from lower pyramid levels (confer b) and c)). The left and right borders have small disparities that arise because those objects have entered the image (from the right) and left if (to the left) such that matches were not possible. In this case the normalized cross correlation was too low and we set the disparity to zero.



(a) Fourth level (b) Third level (c) Second level



(d) First level

Figure 3: Disparity pyramid for Tsukuba

Conclusion

We have seen how to infer depth from two 2D images that are displaced with respect to one another. The best mean error we achieved for Venus was -0.0291. In other words, on average, we were 0.0291 pixels off in our estimation. Tsubuka, which is more complex than Venus, suffers from loss of finer details. This may be partly due to the median filtering. There is a definite trade-off between lower errors (larger patch sizes) and faster computations (smaller patch sizes). The running time increases with the search radius, but not as much as with the patch size. We have also seen that in order for the method to work properly, we need texture rich surfaces with high variation. The Map images have the texture, but since it is the same type of texture all over, the method makes more mistakes than with Venus.

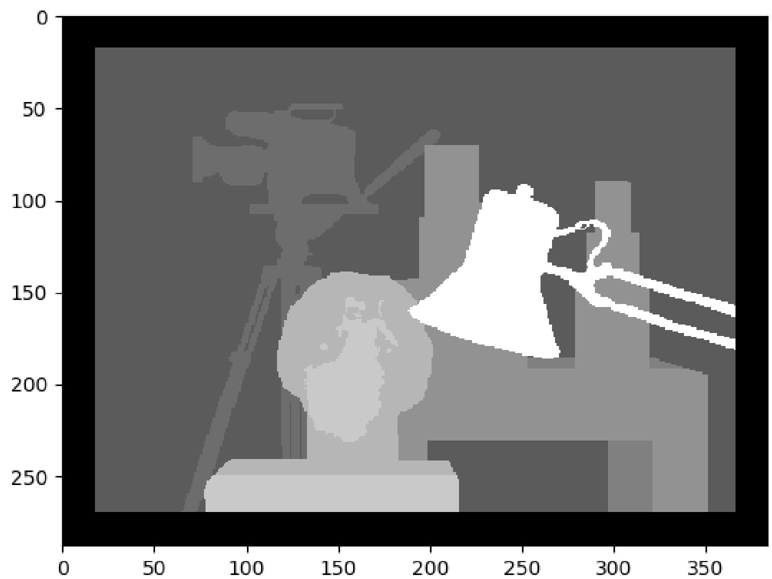


Figure 4: True disparities of Tsukuba